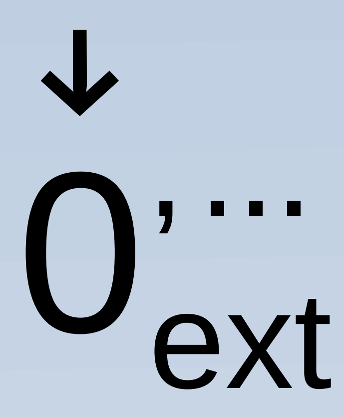


Иван *Futurologic* Петров



Гипернулярион: одно краткое частное представление наименьшего и наибольшего числа во Вселенной.

Содержание данной публикации предназначено для читателей, увлекающихся математическими абстракциями и философскими размышлениями. Рекомендуется для возраста 16+.

2024

Материал, представленный в данной публикации, является результатом творческого труда автора и отражает его личную точку зрения. Автор пришел к описанным идеям самостоятельно, не претендуя на полную оригинальность. Публикация не ставит перед собой научных или просветительских целей, а представляет собой **частную аналитическую работу**. Автор выражает уважение к трудам других авторов и не стремится оспаривать первенство их идей. На момент публикации автору не известны аналогичные работы, описывающие рассматриваемые идеи подобным образом. Содержание публикации формировалось исключительно за счет творческой и интеллектуальной деятельности автора, опираясь только на его личные знания в данной области. Таким образом, автор самостоятельно создал данную публикацию, включая описание и представление идеи. Он выражает готовность уважительно относиться к интеллектуальной собственности других авторов, и в случае обнаружения аналогичных материалов, ранее опубликованных или зарегистрированных другими авторами, все права и приоритеты остаются за ними.

Автор данной публикации не призывает к каким-либо действиям и не стремится оскорбить чувства читателей. Цель материала заключается в выражении личного мнения без намерения вызвать недопонимание или негативные эмоции. Любые совпадения с реальными лицами, именами, понятиями, терминами, событиями, названиями, торговыми марками, брендами, а также с созвучными словами из других языков и публикаций являются случайными, и автор не несет ответственности за возможные недоразумения.

Автор данного материала не несет ответственности за возможные опечатки, неточности, ошибки или неправильную интерпретацию содержания. Вся предоставленная информация представлена "как есть", и читатель самостоятельно несет ответственность за оценку и использование представленных данных.

Автор данной публикации не несет ответственности за возможные последствия использования электронного файла публикации, ознакомления с материалом и его применения на практике. Ответственность за интерпретацию, использование и возможные последствия лежит исключительно на читателе. Автор также не несет ответственности перед читателем или третьими лицами за действия, основанные на содержании данного материала.

Авторский текст данной публикации был откорректирован с использованием искусственного интеллекта. Роль искусственного интеллекта ограничивается лишь вспомогательной функцией — проверкой и исправлением пунктуационных, стилистических и грамматических ошибок. ChatGPT (GPT-3, крупномасштабная модель генерации языка от OpenAI) был частично использован для проверки текста и улучшения стиля написания этой редакционной статьи. Автор рассмотрел все правки, отредактировал предложенные ChatGPT фразы по своему усмотрению, проверил их и принимает на себя окончательную ответственность за содержание данной публикации.

Математика порой может удивлять своей таинственной способностью к созданию чисел, которые не поддаются интуиции. Одним из таких примеров является *гипернулярион* — число, которое мы можем назвать одной из наименьших величин в мире. Но это "наименьшее" число на самом деле гораздо более сложное, чем может показаться на первый взгляд.

Что такое гипернулярион?

Гипернулярион (обозначаемое как $\overset{\downarrow}{0}_{ext}^{,\cdots}$) — это сверхмалое число, получаемое с использованием комбинации таких мощных математических концепций, как функция *Аккермана*, теория *Busy Beaver* и *гипероперации*. Оно стремится к нулю, но с такой скоростью, что невозможно выразить простыми словами или числами.

Как мы получаем гипернулярион?

Чтобы понять, как возникает *гипернулярион*, нужно разобраться в нескольких математических механизмах:

1. **Функция Аккермана** — функция, которая растёт с ужасающей быстротой и используется для создания чисел, значительно превышающих любые стандартные. Она принимает два целых числа и производит такие огромные числа, что в вычислениях уже трудно их представить.
2. **Busy Beaver** — концепция из теории вычислимости, обозначающая наибольшее количество шагов, которое может выполнить Тьюрингова машина до остановки Она приводит к величинам, которые невозможно выразить привычными числами.
3. **Гипероперации** — это серия операций, начиная от обычного умножения и заканчивая тетрацией и выше. В случае *гипернуляриона* мы применяем такие операции, где значения растут с невероятной скоростью.

Формула гипернуляриона

Гипернулярион может быть представлен через рекурсивную формулу:

$$\overset{\downarrow}{0}_{ext}^{,\cdots} = \frac{1}{A(M, P(M-1)) \uparrow^{k} P(M-1)}$$

где:

- $A(M,P(M-1))$ — это функция Аккермана, примененная к значениям M и $P(M-1)$,
- $\uparrow^k$ — гипероперация, которая определяет сверхбыстрое возрастание значений,
- $k=\Sigma(P(M-1))$ — это функция Busy Beaver, которая вычисляет максимальное количество шагов, выполненное Тьюринг-машиной за $P(M-1)$ шагов,
- $P(M-1)$ — это рекурсивно определённая функция, зависящая от предыдущих значений в последовательности,
- $M=\Sigma(P(10))$.

Число $\overset{\downarrow}{0}^{,\dots}_{ext}$ определяется как предел *сверхубывающей* последовательности, основанной на функции $P(n)$, которая сочетает в себе Аккермана, Busy Beaver и гипероперации.

Таким образом, $\overset{\downarrow}{0}^{,\dots}_{ext}$ — это число, которое стремится к нулю настолько быстро, что его точное значение практически невозможно вычислить, однако оно остаётся строго определённым благодаря рекурсивной структуре.

Так как $\overset{\downarrow}{0}^{,\dots}_{ext}$ одно из сверхмалых чисел, тогда $\frac{1}{\overset{\downarrow}{0}^{,\dots}_{ext}}$ - одно из сверхбольших чисел, равное:

$$\frac{1}{\overset{\downarrow}{0}^{,\dots}_{ext}}=A(M,P(M-1))\uparrow^k P(M-1)$$

Определение функции $P(n)$

Функция $P(n)$ строится рекурсивно:

$$P(n)=\begin{cases} 1, \text{если}\, n=0; \\ A(n,P(n-1))\uparrow^{\Sigma(P(n-1))}P(n-1), \text{если}\, n>0 \end{cases}$$

где:

- $A(m,n)$ — это функция Аккермана, которая определяется как:

$$A(m,n)=\begin{cases} n+1, \text{если } m=0; \\ A(m-1,1), \text{если } m>0 \text{ и } n=0; \\ A(m-1,A(m,n-1)), \text{если } m>0 \text{ и } n>0 \end{cases}$$

- $\Sigma(k)$ = максимальное количество символов, напечатанных Тьюринг-машиной за k шагов,
- $\uparrow^k$ — это гипероперация, которая может быть определена через модификацию:

$$a\uparrow^k b=\begin{cases} a^b, \text{если } k=1; \\ a\uparrow^{k-1}(a\uparrow^{k-1}(\dots(a\uparrow^{k-1}(b-1))))_{k-1 \text{ раз}}, \text{если } k>1 \end{cases}$$

Определение M

Для того чтобы M было конкретным, но невероятно большим, мы определяем его как результат вычислений, связанных с функцией Busy Beaver. Таким образом:

$$M=\Sigma(P(10))$$

Здесь мы используем конкретное значение для $n=10$. После того как мы вычислим $P(10)$, мы получим значение M, которое будет результатом функции Busy Beaver для числа $P(10)$. Это будет гигантское число, которое мы будем использовать для расчёта *гипернуляриона*.

Заключение

Число $\overset{\downarrow}{0}_{ext}^{,\dots}$, которое мы построили, является примером сверхмалого числа, вероятно, превосходящего все известные на текущий момент числовые величины, как по сложности вычислений, так и по масштабу роста. Его уникальность заключается в сочетании нескольких мощных математических концепций: функции Аккермана, Busy Beaver и гипероперации. Каждое следующее значение в последовательности $P(n)$ растёт экспоненциально, становясь настолько огромным, что его невозможно выразить стандартными методами. Это открывает новые горизонты для исследования как больших, так и малых чисел в математике, давая нам ключ к пониманию пределов вычислительных возможностей и теоретических конструкций в мире чисел. $\overset{\downarrow}{0}_{ext}^{,\dots}$ — это не просто абстракция, но и увлекательное математическое открытие, которое заставляет задуматься о том, что происходит на границах человеческого воображения и вычислительных сил.

Дополнение к публикации.

Пояснение 1.

Формула для гипероперации $a\uparrow^k b$ в виде:

$$a\uparrow^k b=\begin{cases} a^b, \text{если} k=1; \\ a\uparrow^{k-1}(a\uparrow^{k-1}(\dots(a\uparrow^{k-1}(b-1))))_{k-1\text{раз}}, \text{если} k>1 \end{cases}$$

представляет собой рекурсивное определение гипероперации, где k указывает на уровень гипероперации, а a и b — числа, для которых вычисляется операция.

Пояснение:

1. **Когда** $k=1$:
   - Это обычная степень: a^b.
   - Например, $2\uparrow^1 3=2^3=8$.
2. **Когда** $k>1$:
   - В этом случае мы применяем рекурсивное вычисление *гипероперации* с уменьшением уровня k и операцией, которая опять использует гипероперацию.

Пример для $k=2$**:**

Рассмотрим, например, $a=2$ и $b=3$.

1. Для $k=2$:

   $2\uparrow^2 3=2\uparrow^1(2\uparrow^1(3-1))$

2. Сначала вычислим $3-1=2$, то есть:

   $2\uparrow^2 3=2\uparrow^1(2\uparrow^1 2)$

3. Теперь считаем $2\uparrow^1 2$ — это просто степень:

   $2\uparrow^1 2=2^2=4$

4. Подставляем значение обратно:

   $2\uparrow^2 3=2\uparrow^1 4=2^4=16$

Таким образом: $2\uparrow^2 3=16$.

Пример для $k=3$:

Теперь рассмотрим $k=3$, что уже приведёт к более сложным вычислениям.

1. Для $k=3$:

   $$2\uparrow^3 3=2\uparrow^2\left(2\uparrow^2\left(2\uparrow^2(3-1)\right)\right)$$

2. Сначала вычислим $3-1=2$, то есть:

   $$\left(2\uparrow^2\left(2\uparrow^2(3-1)\right)\right)=2\uparrow^2\left(2\uparrow^2 2\right)$$

3. Теперь вычислим $2\uparrow^2 2$, что уже является терацией:

   $$2\uparrow^2 2=2\uparrow^1\left(2\uparrow^1 1\right)=2\uparrow^1 2=2^2=4$$

4. Теперь вернемся к вычислению:

   $$\left(2\uparrow^2\left(2\uparrow^2(3-1)\right)\right)=2\uparrow^2 4=2\uparrow^1\left(2\uparrow^1 3\right)$$

5. Теперь вычислим $2\uparrow^1 3$, что снова является степенью:

   $$2\uparrow^1 3=2^3=8$$

6. Вычисляем $2\uparrow^2 3$:

   $$\left(2\uparrow^2\left(2\uparrow^2(3-1)\right)\right)=2\uparrow^1 8=2^8=256$$

7. Подставляем обратно:

   $$2\uparrow^3 3=2\uparrow^2 256=2\uparrow^1\left(2\uparrow^1 255\right)$$

8. Вычисляем $2\uparrow^1\left(2\uparrow^1 255\right)$:

   $$2\uparrow^1\left(2\uparrow^1 255\right)=2^{\left(2^{255}\right)}$$

Итак, $2\uparrow^3 3=2^{\left(2^{255}\right)}$, означает, что число 2 возводится в степень 255, затем в результате возводится 2 в степень равную числу, полученному на первом этапе, образуя так называемую "степенную лесенку".

Заключение:

Когда $k>1$, гипероперация требует рекурсивного вычисления, и с каждым увеличением k значения чисел растут экспоненциально быстрее. Пример с $k=2$ и $k=3$ показывает, как быстро нарастают величины, и как сложность вычислений увеличивается с каждым шагом.

Пояснение 2.

Описание функции Busy Beaver (в контексте предложенной модели)

Функция **Busy Beaver** ($\Sigma(n)$) измеряет максимальное количество единичных символов, которое может напечатать Тьюринг-машина, начиная с пустой ленты и работая не более nnn шагов. Важно отметить, что для каждого n существует конечное максимальное значение, которое можно напечатать, но эта величина растёт крайне быстро с увеличением n. Функция **Busy Beaver** — это суперэкспоненциальная функция.

Для конкретного числа n функция $\Sigma(n)$ даёт максимальное количество единичных символов, которые могут быть напечатаны Тьюринг-машиной, которая выполняет не более n шагов и останавливается. Для вычислений функции Busy Beaver используется **универсальная** Тьюринг-машина, которая *начинает с пустой ленты и исходного состояния*.

Формальное определение

Для данного числа n, функция **Busy Beaver** определяется как:

$$\Sigma(n) = \text{максимальное количество единичных символов, напечатанных Тьюринг-машиной за } n \text{ шагов на пустой ленте.}$$

Эти вычисления происходят на основе универсальной Тьюринг-машины, которая начинает с пустой ленты, и после выполнения n шагов, она останавливается. В процессе работы она может только печатать единичные символы (или оставлять пустые клетки) на ленте.

Важно, что функция **Busy Beaver** является **невычислимой**. Это означает, что не существует алгоритма, который бы точно и эффективно вычислял значение $\Sigma(n)$ для всех n, потому что для больших значений n вычисления становятся чрезвычайно сложными.

Реальные значения для $\Sigma(n)$

Для малых значений n функция $\Sigma(n)$ имеет следующие известные значения:

1. **Для $n=1$:**

   Максимальное количество единичных символов, которые могут быть напечатаны Тьюринг-машиной за один шаг:

   $\Sigma(1)=1$

2. **Для** $n=2$ **:**

   За два шага можно напечатать 4 символа (реализуется более сложной Тьюринг-машиной):

   $\Sigma(2)=4$

3. **Для** $n=3$ **:**

   За три шага можно напечатать 6 символов:

   $\Sigma(3)=6$

4. **Для** $n=4$ **:**

   За четыре шага машина может напечатать до 13 единичных символов:

   $\Sigma(4)=13$

5. **Для** $n=5$ **:**

   За пять шагов количество единичных символов ещё увеличивается:

   $\Sigma(5)=4098$

   В этом случае значение $\Sigma(5)$ является *экстраординарно* большим.

6. **Для** $n=6$ **:**

   За шесть шагов можно напечатать уже невероятно большое количество символов:

   $\Sigma(6)\approx 10101034$

   Здесь Σ(7) — это настолько большое число, что оно выходит за пределы стандартных вычислительных возможностей, и его невозможно выразить в обычной десятичной записи.

Для нашего примера

В нашем контексте мы используем **функцию Busy Beaver** для вычисления значения M, которое будет невероятно большим, но вполне определённым. В частности, M будет результатом функции $\Sigma(n)$, где n — это достаточно большое число, выбранное таким образом, чтобы результат был исключительным.

$M=\Sigma(n)$, где n — это число, для которого мы вычисляем величину $\Sigma(n)$.

Например, если мы выбрали $n=P(10)$, то:

$M=\Sigma(P(10))$ — это значение M будет астрономически огромным.

Заключение

Функция **Busy Beaver** — это мощный инструмент, который позволяет строить огромные числа, которые имеют суперэкспоненциальный рост. Для нашего примера $M=\Sigma(n)$, это число будет настолько большим, что его нельзя будет выразить стандартными методами записи чисел.

Пояснение 3.

Пример вычислений для P(n)

- Для $n=0$:

  $P(0)=1$

  Это начальное значение, так как мы задаём базу для рекурсии.

- Для $n=1$:

  $P(1)=A(1,P(0))\uparrow^{\Sigma(P(0))}P(0)=A(1,1)\uparrow^{\Sigma(1)}1=3$

  Здесь:

  $A(1,1)=3$, так как по определению функции Аккермана $A(m,n)$ для $m=1$ и $n=1$ даёт результат 3.

  $\Sigma(1)=1$, так как функция Busy Beaver для $P(0)=1$ даёт 1.

  Таким образом:

  $P(1)=3\uparrow^{1}1=3^{1}=3$

- Для $n=2$:

  $P(2)=A(2,P(1))\uparrow^{\Sigma(P(1))}P(1)=A(2,3)\uparrow^{\Sigma(3)}3$

  Здесь:

  $A(2,2)=7$, так как по определению функции Аккермана $A(2,2)$ даёт 7.

- $\Sigma(3)=6$, так как функция Busy Beaver для $P(1)=3$ даёт 6.

  Таким образом:

  $P(2)=7\uparrow^{6}2$

  Здесь: $7\uparrow^{6}2$ — это уже гипероперация, которая даёт результат, экспоненциально превосходящий обычные числа.

- Для n=3:

  $P(3)=A(3,P(2))\uparrow^{\Sigma(P(2))}P(2)$

  Здесь:

- $A(3,P(2))$ — это уже чрезвычайно сложное вычисление, которое будет иметь результат, сильно превышающий стандартные значения.

- $\Sigma(P(2))$ — функция Busy Beaver для $P(2)$, которая даёт величину, настолько огромную, что её невозможно записать обычным способом.

Таким образом:

$P(3)=A(3,P(2))\uparrow^{\Sigma(P(2))}P(2)$ это число будет настолько большим, что его невозможно выразить стандартными средствами

Каждое следующее значение для $P(n)$ будет стремиться к вычислениям, которые невозможно явно записать из-за огромного роста чисел.